**Министерство образования Республики Беларусь**

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

**ОТЧЕТ**

по лабораторной работе

на тему:

Распознавание и поиск лексем

Выполнил

Студент гр. 051005 **К.Н. Волков**

Проверил **Асс. C.В. Болтак**

Минск, 2021

# 1. Задание

**1.1 Постановка задачи**

# 

# 

# 2.2 Ожидаемый результат

# 

|  |  |
| --- | --- |
| test@localhost | true |
| vasya@pupkin.ru | true |
| ivan.ivanov@mail.bsuir.by | true |
| a.b.c@d-e-f.com | true |
| a@g | true |
| 123@123 | true |
| pri.@com | false |
| pr.i@com-.ru | false |
| pri@co-m.ru | true |
| sasd+-?@mail.ru | false |
| asdasdsadsad | false |
| max2130.sad@a-b-c.com.ru | true |

Строка:

test@localhost asdsad [vasya@pupkin.ru](mailto:vasya@pupkin.ru) ///…@o [a.b.c@d-e-f.com](mailto:a.b.c@d-e-f.com) 123 [ivan.ivanov@mail.bsuir.by](mailto:ivan.ivanov@mail.bsuir.by)

Результат:

test@localhost [vasya@pupkin.ru](mailto:vasya@pupkin.ru) [a.b.c@d-e-f.com](mailto:a.b.c@d-e-f.com) ivan.inanov@mail.bsuir.by

**Регулярное выражение**

**Классы:**

UpLetters = [A-Z];

LowLetters = [a-z];

Digits = [0-9];

Dot = ‘.’;

Sobaka = ‘@’;

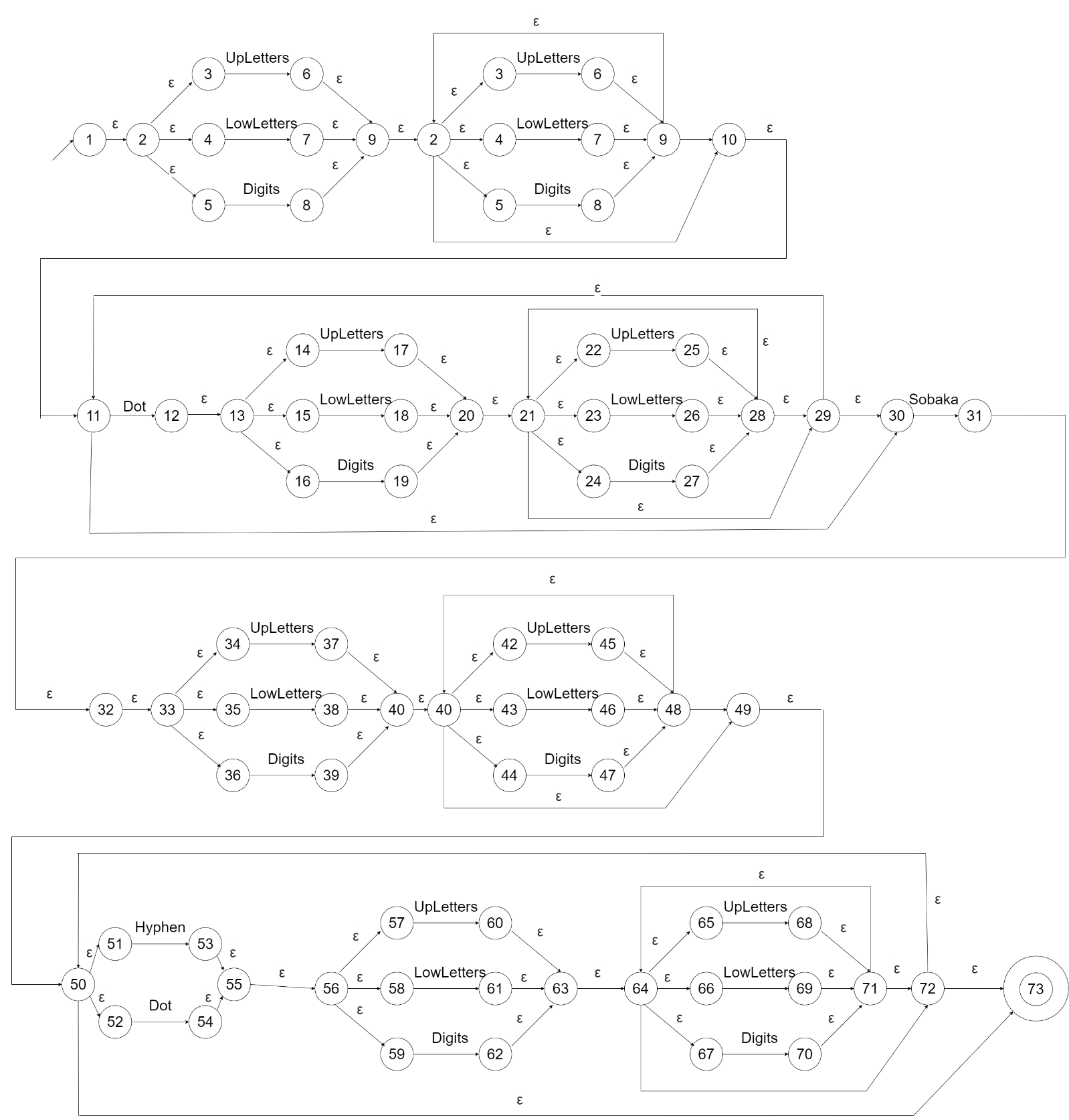
Hyphen = ’-‘;

**Регулярное выражение:**

Mail = (UpLetters|LowLetters|Digits)+ ((Dot (UpLetters|LowLetters|Digits)+)\*) Sobaka

(UpLetters|LowLetters|Digits)+ (((Hyphen|Dot) (UpLetters|LowLetters|Digits)+)\*)

**НКА:**

****

# ДКА:

# 

**Таблица переходов:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | UpLet | LowLet | Dig | Sob | Def | Dot |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 2 | 2 | 2 | \* | \* | \* |
| 2 | 2 | 2 | 2 | 3 | \* | 1 |
| 3 | 4 | 4 | 4 | \* | \* | \* |
| 4 | 4 | 4 | 4 | \* | 3 | 3 |

**3.Выполнение**

# 3.1 Код программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include <locale.h>

#define sep " \n\t"

enum charType {

ctUnknown, //0

ctUpLetters, // 1

ctLowLetters,//2

ctDigits,//3

ctSobaka,//4

ctHyphen,//5

ctDot,//6

};

const int transitions[5][7] = {

{0, 0, 0, 0, 0, 0, 0}, //Ошибка

{0, 2, 2, 2, 0, 0, 0}, //1

{0, 2, 2, 2, 3, 0, 1}, //2

{0, 4, 4, 4, 0, 0, 0}, //3

{0, 4, 4, 4, 0, 3, 3}, //4

};

//vasya@pupkin.ru ivan.ivanov@mail.bsuir.by a.b.c@d-e-f.com фыввы@com volkovkonstantinn@gmail.com s@n

const bool isFinalState[5] = {false, false, false, false, true};

const int SIZE = 100;

charType getCharType(char ch) {

if (ch == 'A' || ch == 'B' || ch == 'C' || ch == 'D' || ch == 'E' || ch == 'F' || ch == 'G' || ch == 'H' || ch == 'I' || ch == 'J' || ch == 'K' || ch == 'L' || ch == 'M' || ch == 'N' || ch == 'O' ||

ch == 'P' || ch == 'Q' || ch == 'R' || ch == 'S' || ch == 'T' || ch == 'U' || ch == 'V' || ch == 'W' || ch == 'X' || ch == 'Y' || ch == 'Z')

return ctUpLetters;

else if (ch == 'a' || ch == 'b' || ch == 'c' || ch == 'd' || ch == 'e' || ch == 'f' || ch == 'g' || ch == 'h' || ch == 'i' || ch == 'j' || ch == 'k' || ch == 'l' || ch == 'm' || ch == 'n' || ch == 'o' ||

ch == 'p' || ch == 'q' || ch == 'r' || ch == 's' || ch == 't' || ch == 'u' || ch == 'v' || ch == 'w' || ch == 'x' || ch == 'y' || ch == 'z')

return ctLowLetters;

else if (ch == '0' || ch == '1' || ch == '2' || ch == '3' || ch == '4' || ch == '5' || ch == '6' || ch == '7' || ch == '8' || ch == '9')

return ctDigits;

else if (ch == '@')

return ctSobaka;

else if (ch == '-')

return ctHyphen;

else if (ch == '.')

return ctDot;

else

return ctUnknown;

}

int getInt() {

int n = 0;

int ind = 0;

do {

rewind(stdin);

ind = scanf\_s("%d", &n);

if (ind != 1) {

printf("Ошибка ввода. Пожалуйста, повторите попытку: ");

}

} while (ind != 1);

return n;

}

int splitString(char str[SIZE], char\* words[SIZE]) {

bool isCorrect;

char\* word;

int i = 0;

for (word = strtok(str, sep); word != NULL; word = strtok(NULL, sep)) {

words[i] = word;

i++;

}

return i;

}

bool checkString(char\* word) {

int state = 1;

int i = 0;

for (i = 0; i < strlen(word); i++) {

state = transitions[state][getCharType(word[i])];

}

return isFinalState[state];

}

void printSubStrings(char\* words[SIZE], int numWords) {

bool isCorrect = true;

for (int i = 0; i < numWords; i++) {

if (checkString(words[i])) {

isCorrect = false;

printf("%s ", words[i]);

}

}

if (isCorrect) {

printf("\nКорректные строки отсутствуют\n");

}

}

int main()

{

setlocale(LC\_ALL, "RUS");

char str[SIZE] = { 0 };

printf("Введите строку: ");

fgets(str, 100, stdin);

printf("------------------------------------------\n");

rewind(stdin);

printf("1)Проверка строки\n2)Найти подстроки\nВвод: ");

int num = getInt();

printf("------------------------------------------\n");

if (\*str != '\n') {

if (num == 1) {

if (checkString(str)) {

printf("Строка корректна\n");

}

else {

printf("Строка некорректна\n");

}

}

else if (num == 2) {

char\* words[SIZE] = { 0 };

int numWords = splitString(str, words);

printf("Корректные подстроки:\n");

printSubStrings(words, numWords);

printf("\n");

}

else {

printf("Выход");

}

}

else {

printf("\nСтрока пуста\n");

}

}

**4.Скриншоты**